

DISEÑO LÓGICO

Contenido

INTRODUCCIÓN	2
CÓMO ABORDAR EL DISEÑO	3
TRANSFORMACIÓN DE LAS RELACIONES BINARIAS	4
BINARIAS 1-1 y 1-N SIN ATRIBUTOS	4
BINARIAS 1-1 y 1-N CON ATRIBUTOS	
BINARIOS NN	12
TRANSFORMACIÓN DE LAS REFLEJIVAS.....	13
TRANSFORMACIÓN DE LAS DÉBILES.....	14
TRANSFORMACIÓN DE LAS ESPECIALIZACIONES	15
TRANSFORMACIÓN DE LAS TERNARIAS	16
Con Cardinalidades mínimas individuales de1.....	19
TRANSFORMACIÓN DE LAS AGREGACIONES	19

INTRODUCCIÓN

Modelo Entidad Relación o modelo es

El modelo ER es un modelo de datos que tiene como resultado un diagrama ER o diagrama Chen donde gráficamente se pueden identificar los principales elementos de datos, sus características más importantes y las interrelaciones entre éstos

Fases del diseño de BD

Las fases del diseño de BD son:

1. Diseño conceptual
2. Diseño lógico
3. Diseño físico

El proceso de obtención de un esquema relacional o lógico que represente adecuadamente todos los aspectos estáticos expresados en el esquema conceptual (descritos en el diagrama Entidad-Relación y en el conjunto de restricciones de integridad añadidas) consiste en transformar el diagrama entidad-relación en un esquema relacional aplicando una serie de reglas que, dependiendo del objeto a transformar, constará de un conjunto de relaciones o tablas que lo representan adecuadamente.

También deben ser tratadas las posibles restricciones de el esquema conceptual (RSC) traduciéndolos en expresiones equivalentes o integrándolas en con restricciones del modelo s. En otros casos, aparecen nuevas restricciones de integridad en el esquema relacional.

En algunos casos puede suceder que haya varios esquemas relacionales que representan los requisitos de un esquema conceptual pero sólo uno es el óptimo. El criterio de elección que se aplicará será lo siguiente:

1. Esquema con menos restricciones de integridad de usuario.
2. Ante igualdad de restricciones de usuario, esquema con menor número de relaciones.

Para estudiar estas transformaciones, estudiaremos posibles estructuras del diagrama entidad-relación, presentándose en cada caso el conjunto de relaciones y restricciones equivalentes.

CÓMO ABORDAR EL DISEÑO

Es conveniente seguir un cierto orden a la hora de diseñar lógicamente una base de datos. Una buena práctica puede consistir en proceder de la siguiente manera:

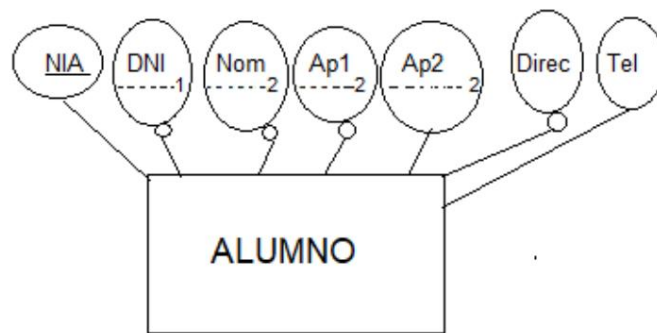
1. En primer lugar, es necesario transformar las entidades del diagrama ER con el que trabajamos en relaciones (mesas).

Cada entidad dará origen a una mesa y los atributos de la entidad serán los de la mesa. En el caso de las fuertes, la clave primaria estará formada por los atributos que forman el identificador de la entidad.

El caso de las débiles, lo abordaremos más adelante.

También habrá que reflejar las restricciones sobre los atributos de la entidad: los atributos obligatorios pasarán a VNN en la mesa (a excepción de los cuales está en la CP que ya lo es implícitamente) y las unicidades en la entidad también lo serán en la mesa.

Ejemplo:



ESTUDIANTE (NIA, DNI, Nombre, Ap1, Ap2, Dirección, Tel)

CP:{NIA}

VNN:{DNI, Nom, Ap1, Direc}

UNI:{DÍAS}

UNI:{Nombre, Ap1, Ap2}

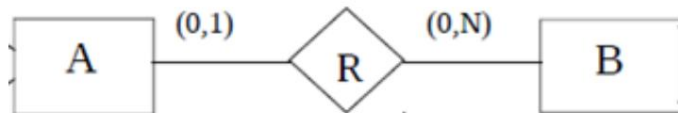
2. Luego se debe seguir transformando en relaciones las entidades que presentan alguno tipos de especificidad (es decir, las débiles, las asociativas, o las derivadas de un proceso de generalización o especialización).
3. A continuación, deben añadirse a las anteriores relaciones los atributos necesarios para formar las claves foráneas derivadas de las interrelaciones binarias con conectividad máxima 1-1 y 1-N presentes en el diagrama ER. En caso de que la relación binaria no presente atributos, éstas relaciones siempre podrán reflejarse en una de las tablas derivadas de las entidades

DISEÑO LÓGICO

4. Y, finalmente, ya puede empezar la transformación de las interrelaciones binarias con conectividad MN y de las interrelaciones n-arias.

TRANSFORMACIÓN DE LAS RELACIONES BINARIAS

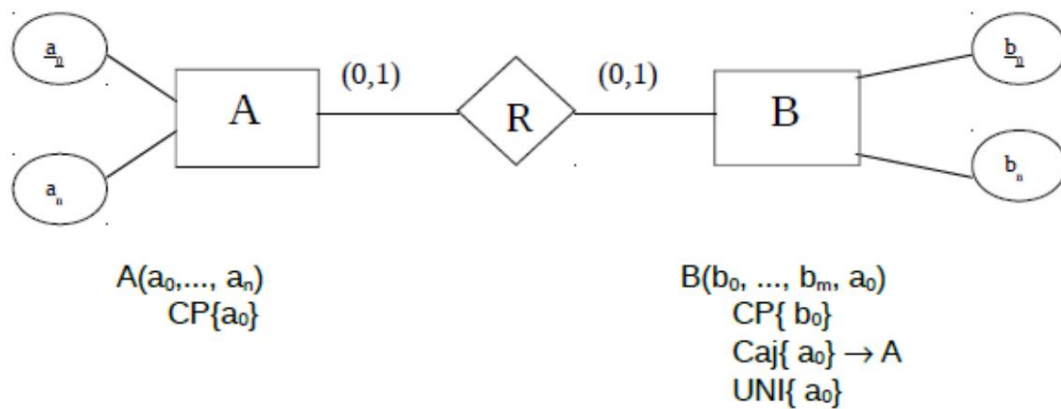
NOTA: Cuando nos referimos a cardinalidad 1-N, por ejemplo, nos estamos refiriendo a las cardinalidades máximas de la relación binaria en el diagrama ER.



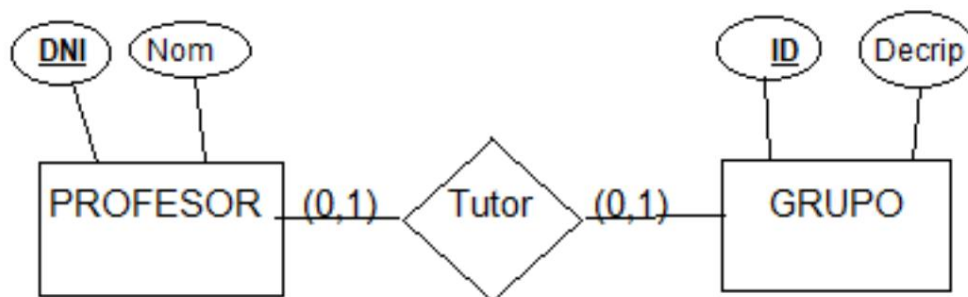
BINARIAS 1-1 y 1-N SIN ATRIBUTOS

En estos casos siempre se puede expresar la relación junto a una de las tablas derivadas de las entidades (la que como máximo participa una vez en la relación).

Caso 1



Ejemplo:



PROFESOR (DÍAS, Nom)

$CP\{DNI\}$

GRUPO (DNI, Decrip, DNI_PROF)

$CP\{Identificación\}$

$UNI\{DNI_PROF\}$

$CAj\{DNI_PROF\} \rightarrow PROFESOR$

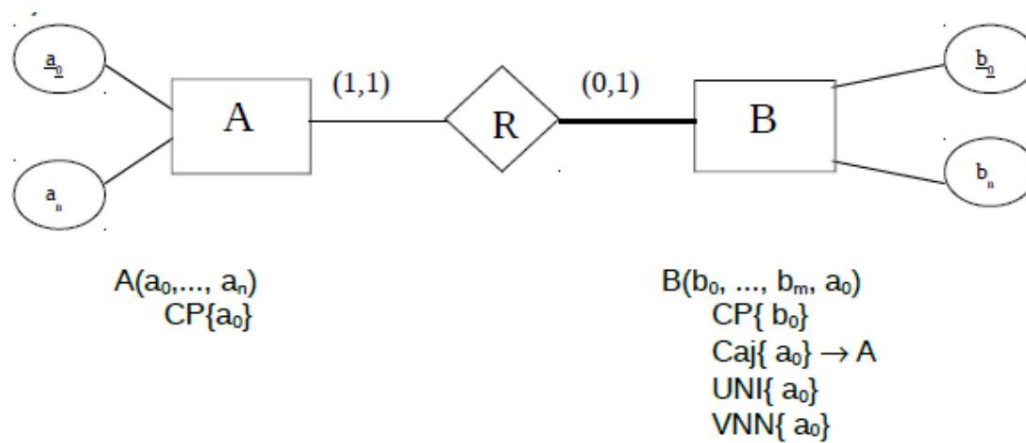
DISEÑO LÓGICO

Con la restricción UNI: {DNI_PROF} nos aseguramos de que un Profesor solo pueda ser Tutor de un Grupo.

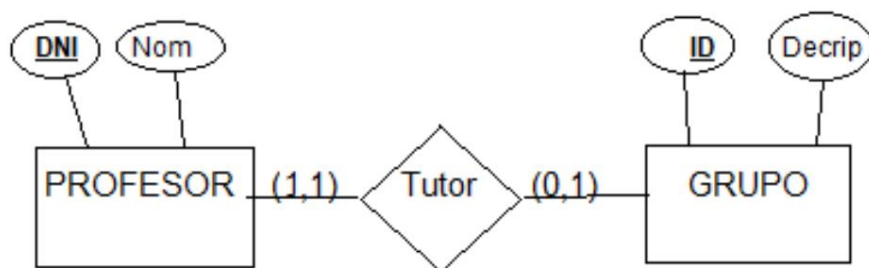
Por supuesto, al ser una cardinalidad simétrica, podíamos haber optado por representar TUTOR en la mesa PROFESOR.

Caso 2

Aquí todas las ocurrencias de B están obligadas a participar en la relación R. Por tanto, la relación debe expresarse en la tabla B; no es como en el Caso 1 que podríamos haber elegido entre expresarla en la tabla A o en la tabla B sean simétricos.



Ejemplo:



PROFESOR (DÍAS, Nom)

$CP\{DNI\}$

GRUPO (DNI, Decrip, DNI_PROF)

$CP\{Identificación\}$

$UNI\{DNI_PROF\}$

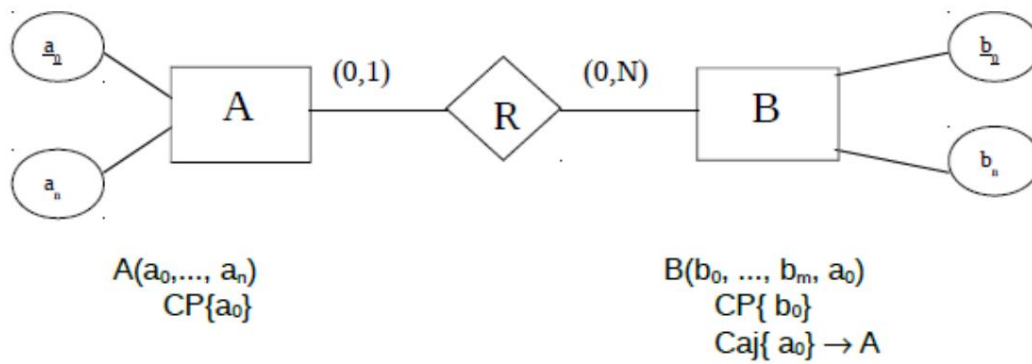
$CAj\{DNI_PROF\} \rightarrow PROFESOR$

$ES\{DNI_PROF\}$

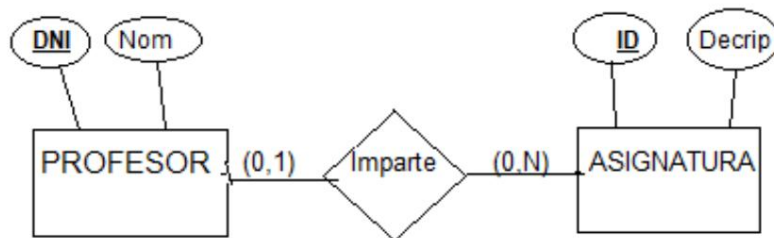
Explicación: Con la restricción de VNN: {DNI_PROF} nos aseguramos que todo GRUPO tenga un TUTOR

DISEÑO LÓGICO

Caso 3



Ejemplo



PROFESOR (DÍAS, Nom)

$CP\{DNI\}$

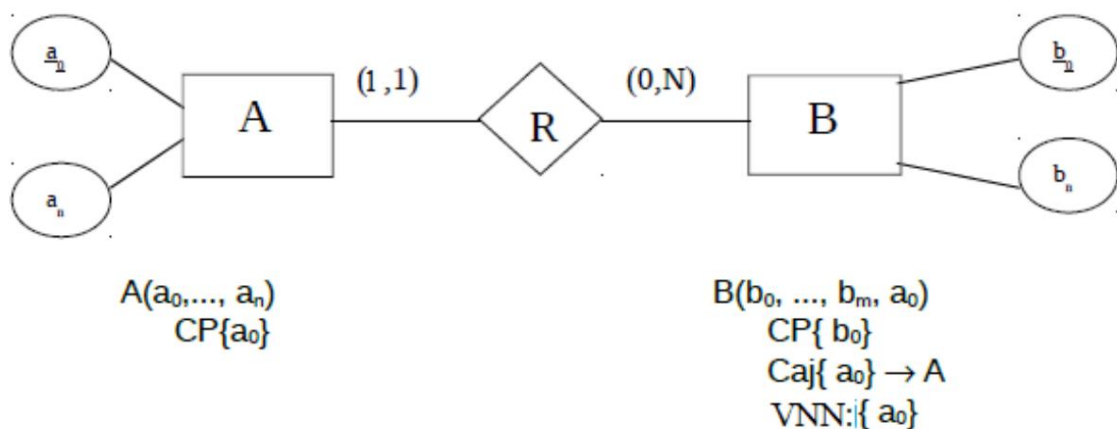
ASIGNATURA (DNI, Decrip, DNI_PROF)

$CP\{Identificación\}$

$CAj\{DNI_PROF\} \rightarrow PROFESOR$

En este caso, como un mismo profesor puede participar varias veces en la relación Imparte, no hay unicidad sobre la Llave Aliena

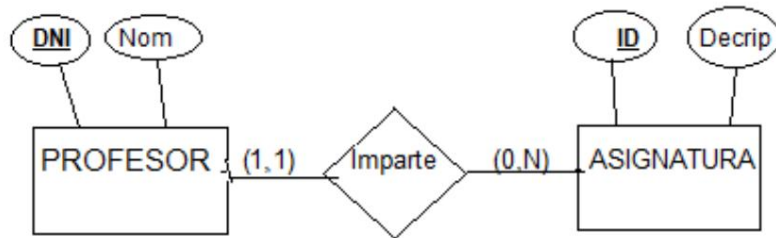
Caso 4



DISEÑO LÓGICO

La obligatoriedad de participación de las ocurrencias de B se consigue con el VNN: {a0}. Cada vez que introduzcamos una fila en la tabla B, obligatoriamente deberemos introducir una referencia a la taula A.

Ejemplo



PROFESOR (DÍAS, Nom)

ASIGNATURA(DNI, Decrip, DNI_PROF)

CP:{DNI}

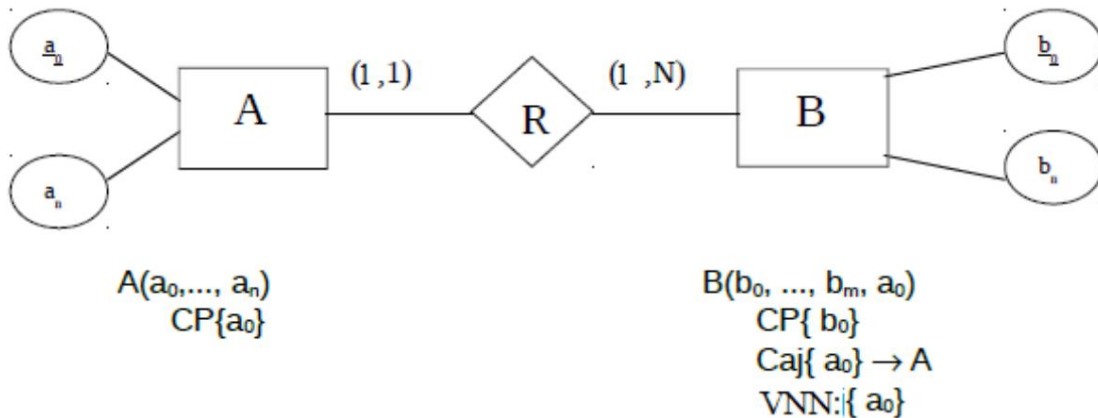
CP:{Identificación}

CAj:{DNI_PROF}->PROFESOR

ES:{DNI_PROF}

Caso 5 (Doble obligatoriedad)

La doble obligatoriedad da lugar a una restricción de usuario si la relación se expresa en una de las tablas derivadas de las entidades; oa dos restricciones en caso de que deba expresarse la relación en una taula diferenciada.



Restricción de obligatoriedad expresada en Cálculo Relacional de Tuplas y SQL respectivamente:

$Ax: A, Bx: B$

$\forall Ax (A(Ax) \rightarrow \exists Bx (B(Bx) \wedge Bx.a_0 = Ax.a_0))$

CREATE ASSERTION Rest_existencia CHECK NOT EXISTS
(SELECT * FROM A WHERE NOT EXISTS
(SELECT * FROM B WHERE B.a0 = A.a0))

DISEÑO LÓGICO

Restricción de obligatoriedad de expresada en Lenguaje Natural:

Toda tupla de la relación (tabla) A está referenciada por al menos una tupla de la relación (tabla) B

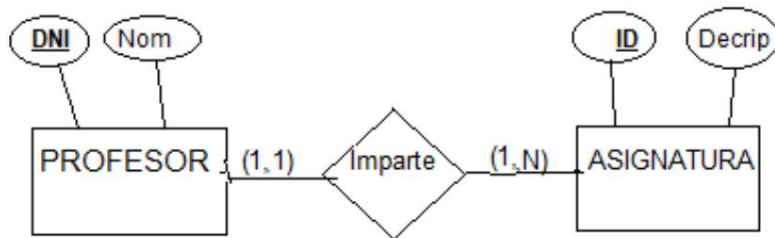
Con la restricción de VNN conseguimos expresar la obligatoriedad de participación de las ocurrencias de B; pero la obligatoriedad de las ocurrencias de A no es posible expresarla mediante restricciones de mesa por lo que nos vemos obligados a introducir una restricción de usuario.

IMPORTANTE

Las expresiones en Cálculo Relacional de Tuplas y SQL NO son contenidos del módulo. Están en de manera ilustrativa.

Las restricciones de usuario se expresarán en lenguaje natural.

Ejemplo:



PROFESOR (DÍAS, Nom)

CP:{DNI}

ASIGNATURA(DNI, Decrip, DNI_PROF)

CP:{Identificación}

CAj:{DNI_PROF}->PROFESOR

ES:{DNI_PROF}

Restricción OBLIGATORIEDAD_PROFESOR: Toda fila de la tabla PROFESOR está referenciada por al menos una fila de la mesa ASIGNATURA.

BINARIAS 1-1 y 1-N CON ATRIBUTOS

El hecho de que una relación tenga atributos propios puede obligar a crear una mesa para expresarla.

Los atributos de una relación se expresan en la tabla en la que está expresada la relación. Pueden suceder dos cosas:

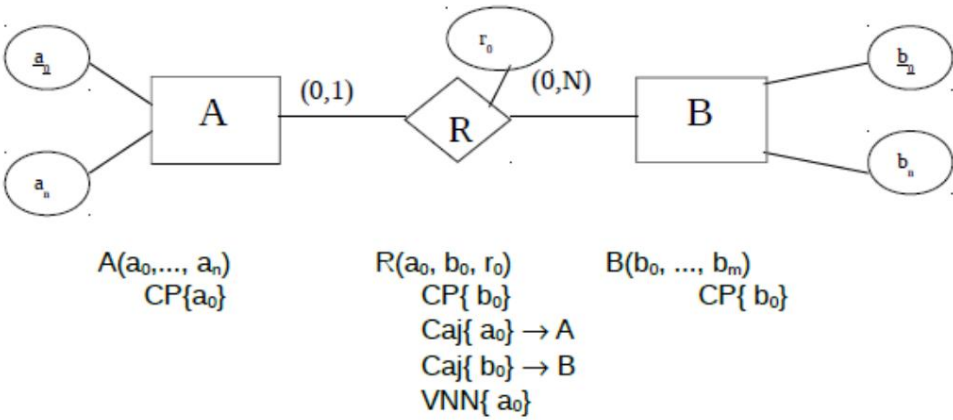
- La clave ajena es obligatoria (todas las filas de la tabla están obligadas a “relacionarse”).

En tal caso el esquema resultante no variará tenga atributo o no la relación.

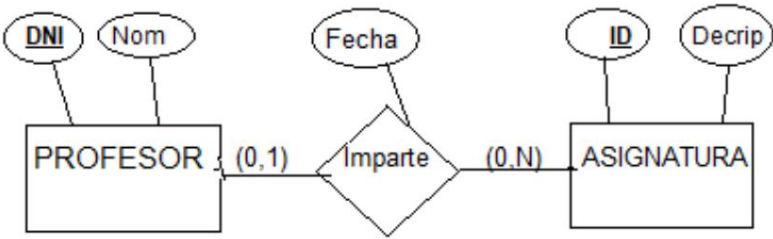
DISEÑO LÓGICO

- La clave ajena es opcional. En este caso se tendrá que construir una mesa aparte para expresar la relación porque al no estar todas las filas relacionadas se podría dar la incongruencia de que una que no estuviera relacionada tuviera valor en el atributo propio de la relación. Una posible (MALA) solución sería introducir una restricción de usuario. Pero, el mejor esquema es el que menos restricciones de usuario tiene. Es preferible introducir una nueva tabla a tener que añadir una restricción.

Caso 6



Ejemplo



MAL DISEÑO.

PROFESOR (DÍAS, Nom)

 $CP\{DNI\}$

ASIGNATURA(ID, Decrip, DNI_PROF, Fecha)

 $CP\{Identificación\}$

$Caj\{DNI_PROF\} \rightarrow PROFESOR$

Restricción Valor_Fecha: En la tabla ASIGNATURA, si el valor del atributo DNI_PROF es NULL, el valor del atributo Fecha debe ser NULL.

DISEÑO LÓGICO

Es decir, con la restricción controlaríamos que si una asignatura no tiene profesor tampoco tenga Data en que empezó a impartirla el profesor.

Aunque añadiendo la restricción de usuario (Restricción Valor_Data) se cumplen todos los requisitos expresados en el modelo conceptual del ejemplo, el diseño no sería el adecuado ya que existe una alternativa sin la restricción de usuario.

BUEN DISEÑO

PROFESOR (DÍAS, Nom)

CP:{DNI}

ASUNTO(ID, Decrip)

CP:{Identificación}

IMPARTE(ID, DNI_PROF, Fecha)

CP:{Identificación}

CAJ:{DNI_PROF}->PROFESOR

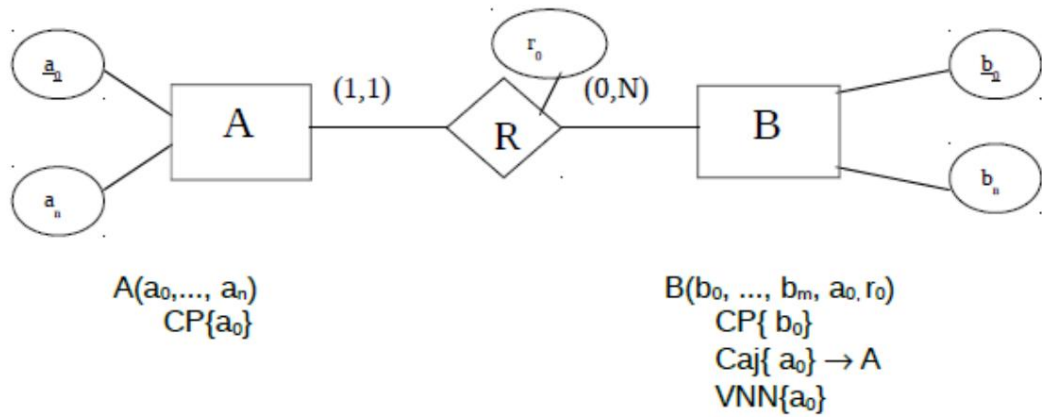
CAJ:{ID}->FIRMA

ES:{DNI_PROF}

EXPLICACIÓN: La Clave Primaria de IMPARTE es la misma que la Clave Aliena dirigida a ASIGNATURA porque asignatura sólo participa una vez en la relación (no se puede repetir).

Por otra parte, IMPARTE representa ocurrencias de la relación. Razón por la que las dos Claves ajenas son obligatorias. Ya hemos asegurado ID haciéndolo Clave Primaria; también es obligatorio DNI_PROF mediante la restricción de VNN.

Caso 7

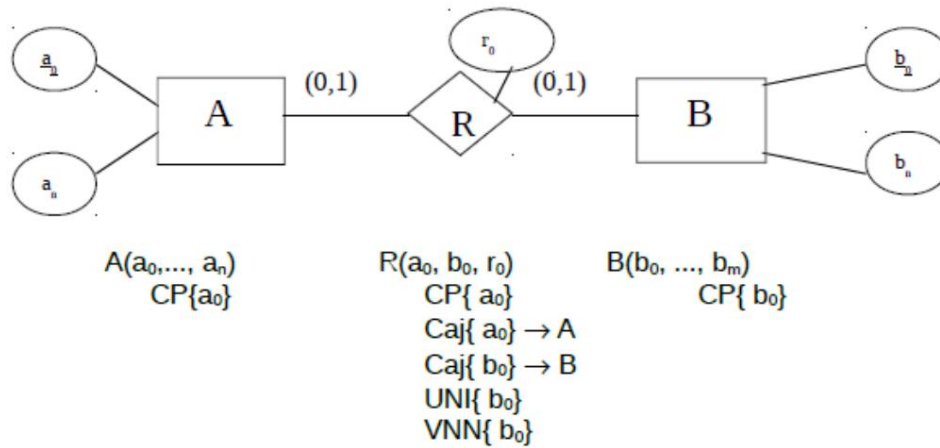


El esquema es el mismo que en el Caso 4

DISEÑO LÓGICO

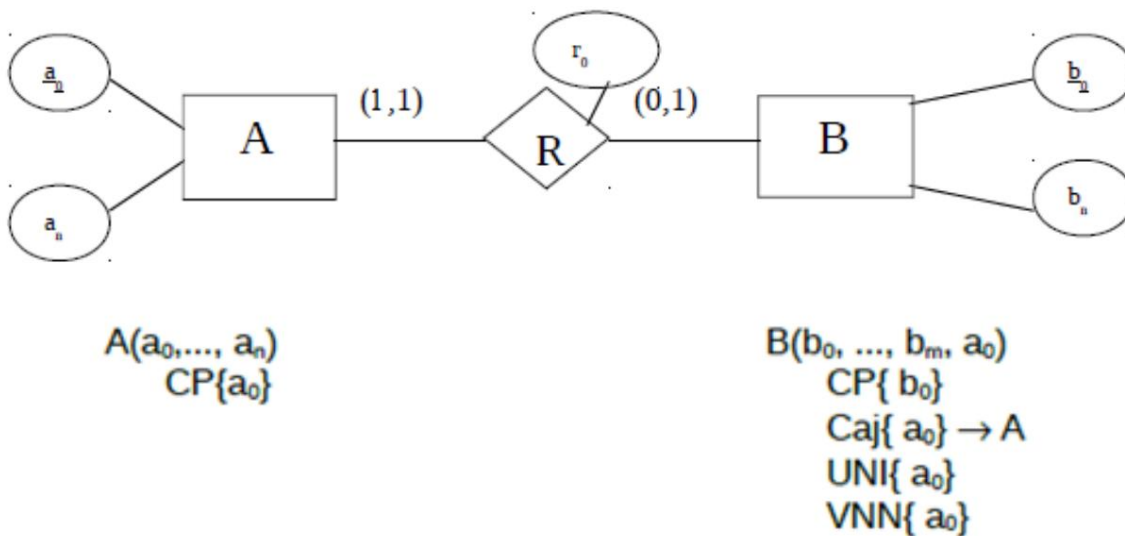
Al estar la relación R representada en la tabla B y ser obligatorio para todas las filas de B estar relacionadas con una de A, el esquema no varía para que R tenga atributos. No puede darse el caso que a_0 sea nulo y por tanto no puede darse la incongruencia.

Caso 8



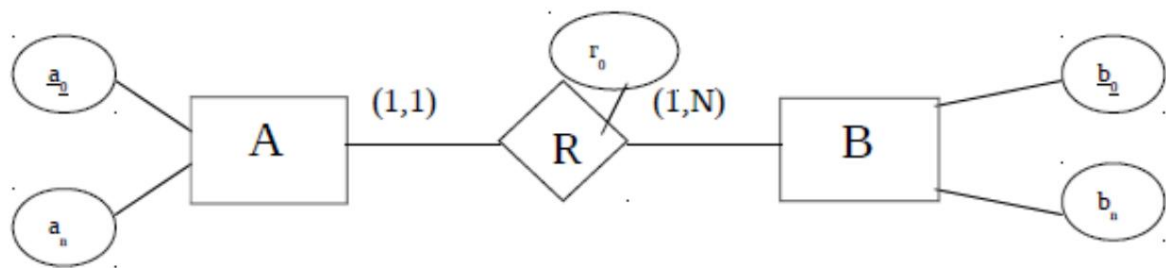
En este caso, el atributo de la relación R determina que deba introducirse una nueva tabla para evitar la restricción de usuario.

Caso 9



Aquí, el esquema es el mismo que en el Caso 2 puesto que no se puede dar el caso de que una fila de B tenga nula la Llave Aliena (a_0); por tanto, no hace falta una restricción para controlar la incongruencia.

Caso 10



$A(a_0, \dots, a_n)$
 $CP\{a_0\}$

$B(b_0, \dots, b_m, a_0, r_0)$
 $CP\{b_0\}$
 $Caj\{a_0\} \rightarrow A$
 $VNN\{a_0\}$

$Ax: A, Bx: B$

$\forall Ax (A(Ax) \rightarrow \exists Bx (B(Bx) \wedge Bx.a_0 = Ax.a_0))$

CREATE ASSERTION Rest_existencia CHECK NOT EXISTS
 (SELECT * FROM A WHERE NOT EXISTS
 (SELECT * FROM B WHERE B.a₀ = A.a₀))

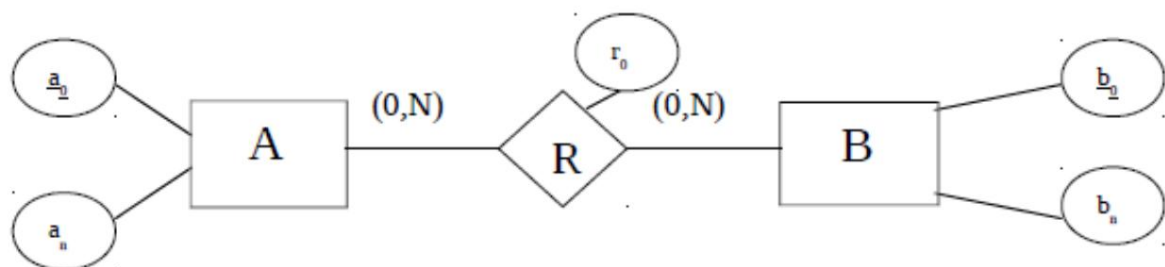
Restricción OBLIGATORIEDAD_A: Toda tupla de la relación (tabla) A está referenciada por al menos una fila de la taula B.

BINARIOS NN

Para todas las binarias con cardinalidades máximas NN el esquema es el mismo: La relación será necesaria expresarla en una mesa separada.

Si alguna de las cardinalidades mínimas es 1 (obligación de participar) habrá que poner una restricción de usuario por cada obligatoriedad de participación.

Caso 11

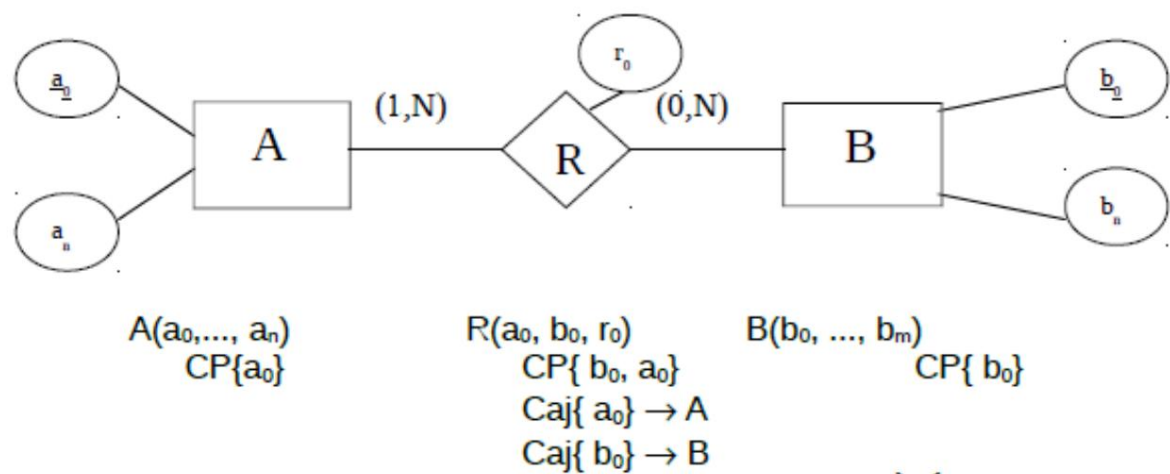


$A(a_0, \dots, a_n)$
 $CP\{a_0\}$

$R(a_0, b_0, r_0)$
 $CP\{b_0, a_0\}$
 $Caj\{a_0\} \rightarrow A$
 $Caj\{b_0\} \rightarrow B$

$B(b_0, \dots, b_m)$
 $CP\{b_0\}$

Caso 12



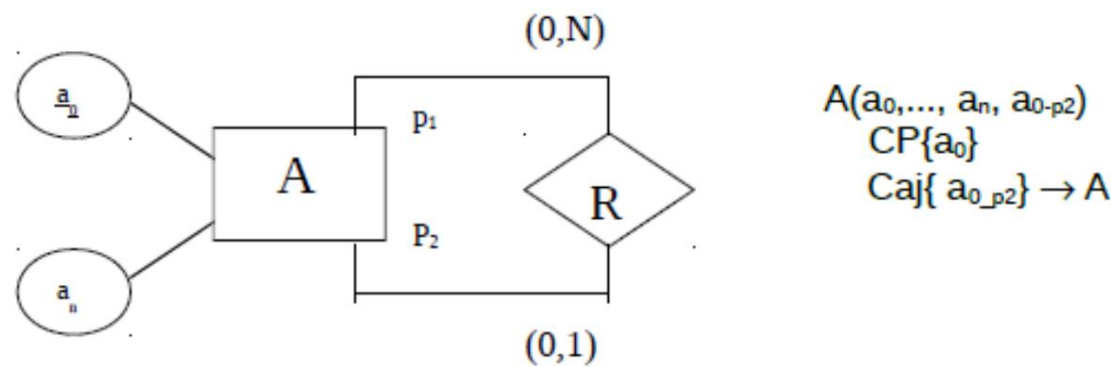
Restricción OBLIGATORIEDAD_B: Toda tupla de la tabla B está referenciada por al menos una fila de la taula R.

TRANSFORMACIÓN DE LAS REFLEJIVAS

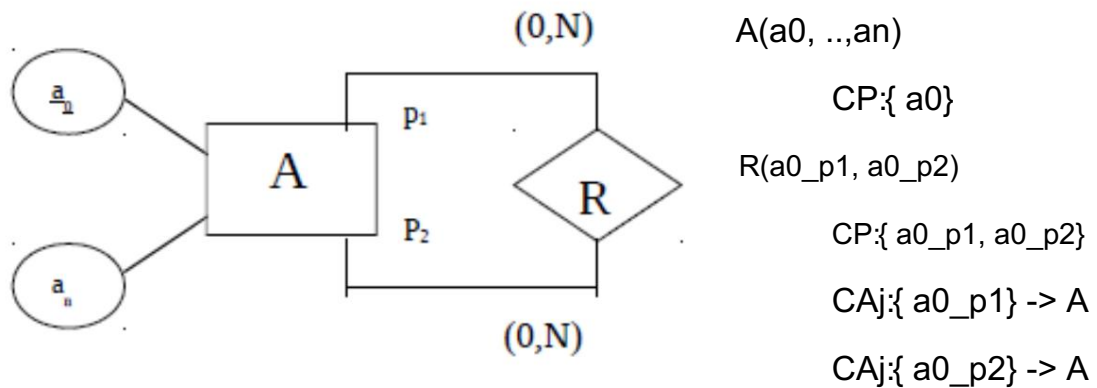
El esquema resultante sigue la misma lógica que el de una binaria puesto que una reflexiva se puede considerar como una relación binaria entre dos copias de una misma entidad.

Las cardinalidades mínimas siempre serán 0 por coherencia (si no, sería como el problema de qué es antes la gallina o el huevo).

Caso 13



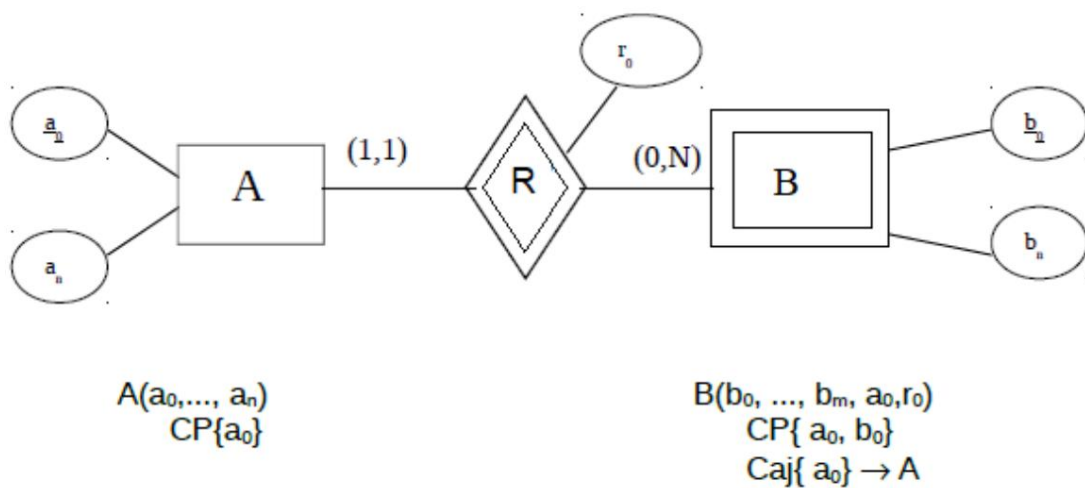
Caso 14



TRANSFORMACIÓN DE LAS DÉBILES

Las ocurrencias de las débiles por identificación necesitan para completar su identificación de su relación con otra entidad. Por tanto, la tabla derivada tendrá una clave primaria formada por el semiidentificador de la débil más el identificador de la fuerte.

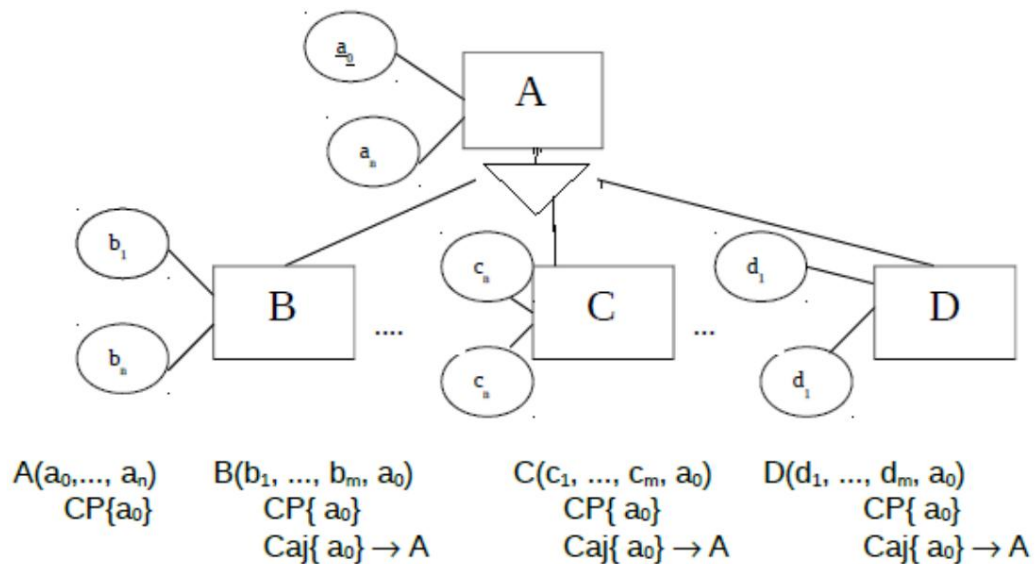
Caso 15



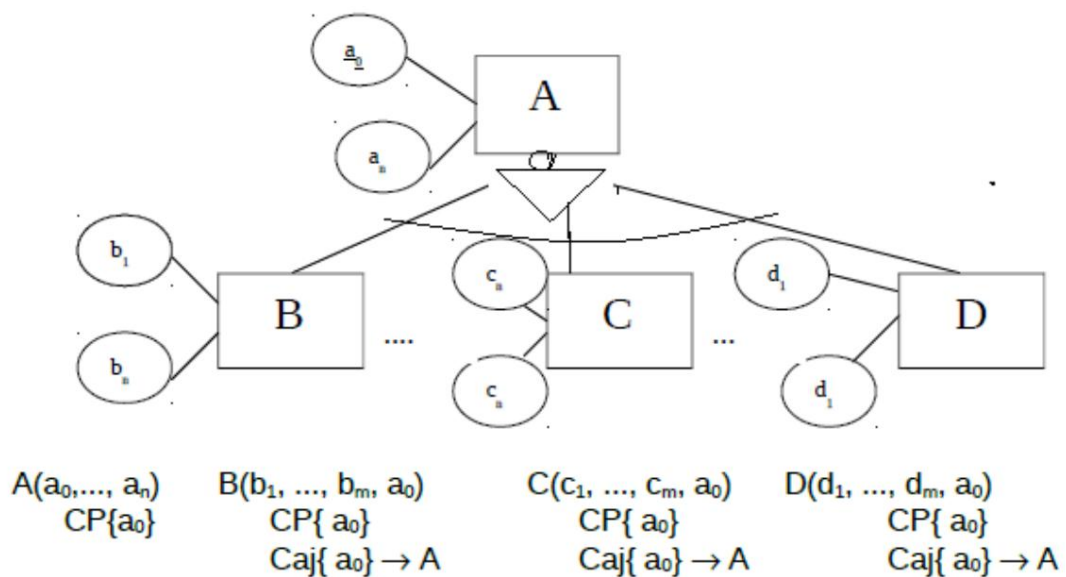
TRANSFORMACIÓN DE LAS ESPECIALIZACIONES

Las especializaciones siempre dan el mismo esquema de tablas; pero habrá que añadir una restricción de usuario si la especialización es TOTAL y otra restricción si es DISJUNTA

Cas 16 (Parcial i Solapada)



Cas 17 (Total i Disjunta)



Restricción total: cualquier tupla de está referenciada por al menos una tupla de B o una tupla de C o una tupla de D.

Restricción Disjunta: No puede haber intersección entre los valores de CP de las tablas B, C y D.

Expresión de las restricciones anteriores en Cálculo Relacional de Tuplas y SQL respectivamente:

TOTAL:

Cálculo relacional de tuplas

$Ax: A, Bx: B, Cx: C, Dx: D$

$$\forall Ax (A(Ax) \rightarrow (\exists Bx (B(Bx) \wedge Bx.a_0 = Ax.a_0) \vee \\ (\exists Cx (C(Cx) \wedge Cx.a_0 = Ax.a_0) \vee \\ (\exists Dx (D(Dx) \wedge Dx.a_0 = Ax.a_0)) \vee$$

SQL

```
CREATE ASSERTION Total CHECK
  NOT EXISTS (SELECT A.a0 FROM A
    WHERE A.a0 NOT IN (SELECT B.a0 FROM B UNION
      SELECT C.a0 FROM C
    UNION
      SELECT D.a0 FROM D ))
```

DESUNIDO:

Cálculo relacional de tuplas

$Bx: B, Cx: C, Dx: D$

$$\exists Bx \exists Cx \exists Dx (B(Bx) \wedge C(Cx) \wedge D(Dx) \wedge \\ (Bx.a_0 = Cx.a_0 \vee Cx.a_0 = Dx.a_0 \vee Dx.a_0 = Bx.a_0))$$

SQL

```
CREATE ASSERTION Disjunta CHECK
  NOT EXISTS (SELECT * FROM B, C, D
    WHERE B.a0 = C.a0 OR B.a0 = D.a0 OR D.a0 = C.a0)
```

TRANSFORMACIÓN DE LAS TERNARIAS

En todas las relaciones ternarias debe añadirse una tabla para representar la relación. Esta mesa tendrá, como es natural, tres claves ajenas: una en cada una de las tablas derivadas de las entidades implicadas y todas serán obligatorias.

Dependiendo de las cardinalidades máximas por parejas, tendremos más o menos unicidades. Cada cardinalidad máxima de 1 dará lugar a una unicidad formada por los atributos representantes de las tablas implicadas.

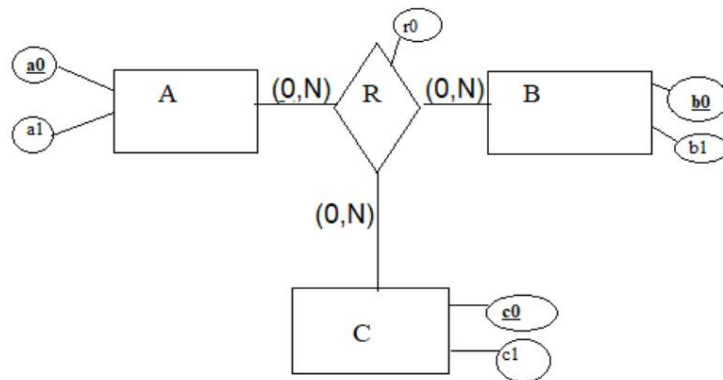
Como puede consultar en las anotaciones del MODELO RELACIONAL las unicidades (o claves) deben ser mínimas. Por tanto, si hay alguna cardinalidad máxima de 1, ya no tendrá sentido la unicidad

DISEÑO LÓGICO

formada por la suma de las tres claves ajenas ya que la formada por la suma de dos de las claves ajenas será un subconjunto de ésta.

Una de estas unicidades (si las hubiere), es necesario promoverla como clave primaria y el resto señalarla las. Se elegirá la que menor número de atributos tenga o cualquiera de ellas si tienen el mismo número de atributos.

Caso 18



A(a0,a1)

CP:{a0}

B(b0, b1)

CP:{b0}

C(c0,c1)

CP:{c0}

R(a0, b0, c0, r0)

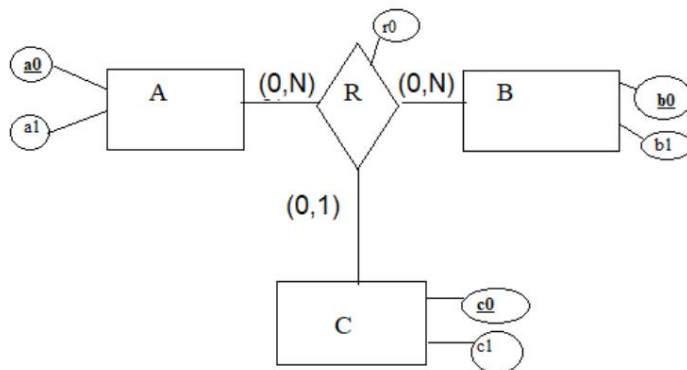
CP:{ a0, b0, c0}

CAj:{a0}->A

CAj:{b0}->B

CAj:{ c0}->C

Caso 19



A(a0,a1)

CP:{a0}

B(b0, b1)

CP:{b0}

C(c0,c1)

CP:{c0}

R(a0, b0, c0, r0)

CP:{a0, b0}

CAj:{a0}->A

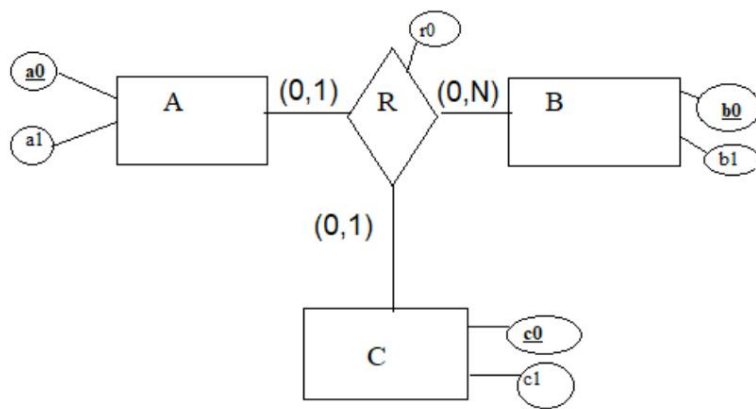
CAj:{b0}->B

CAj:{ c0}->C

ES:{c0}

DISEÑO LÓGICO

Caso 20



A(a0,a1)

CP:{a0}

B(b0, b1)

CP:{b0}

C(c0,c1)

CP:{c0}

R(a0, b0, c0, r0)

CP:{a0, b0}

UNI:{b0,c0}

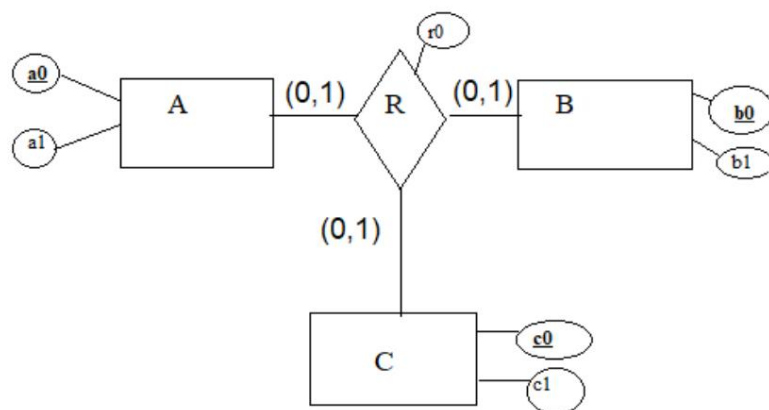
CAj:{ a0} ->A

CAj:{ b0} ->B

CAj:{ c0} ->C

ES:{c0}

Caso 21



A(a0,a1)

CP:{a0}

B(b0, b1)

CP:{b0}

C(c0,c1)

CP:{ c0 }

R(a0, b0, c0, r0)

CP:{a0, b0}

UNI:{b0,c0}

UNI:{a0,c0}

CAj:{ a0} ->A

CAj:{ b0} ->B

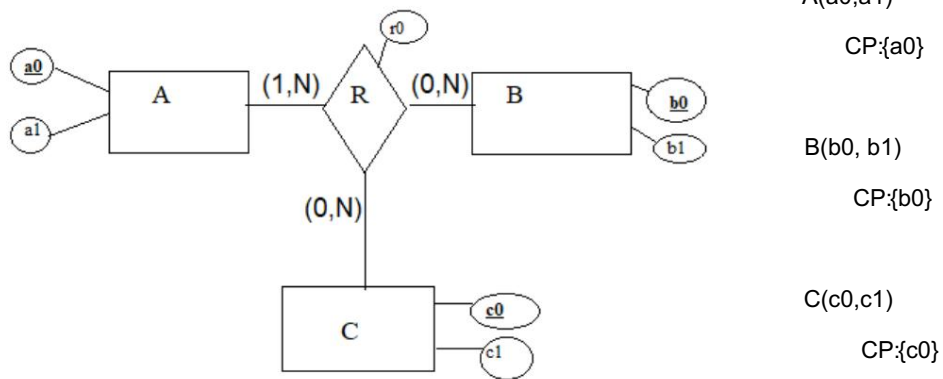
CAj:{ c0} ->C

ES:{c0}

Con Cardinalidades mínimas individuales de 1

Cada una de las cardinalidades mínimas de 1 (obligatoriedad de todas las ocurrencias de una entidad de participar en la relación), dará lugar a una RESTRICCIÓN DE USUARIO para poder representarla.

Caso 22



$R(a_0, b_0, c_0, r_0)$

CP: {a0, b0, c0}

CAj: {a0} → A

CAj: {b0} → B

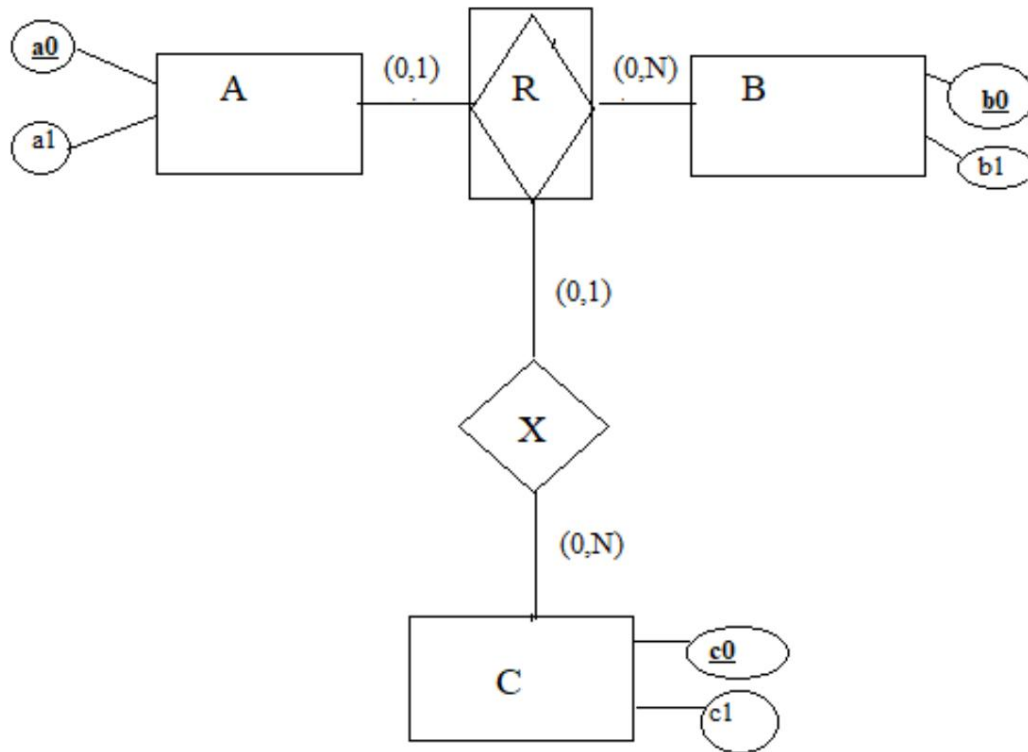
CAj: {c0} → C

RESTRICCIÓN OBLIGATORIEDAD_A: Toda fila de la tabla A está referenciada por al menos una fila de la taula R.

TRANSFORMACIÓN DE LAS AGREGACIONES

Se seguirán los siguientes pasos:

1. Lo primero es transformar la relación binaria que da lugar a la agregación. El esquema resultante será lo mismo que daría si fuera una binaria con esa cardinalidad y atributos en la relación.
2. Una vez tenemos representada la agregación en una de las tablas, actuamos en consecuencia con la cardinalidad y los posibles atributos que existan en la relación entre la agregación y la tercera entidad.



1. Si R (el objeto agregado) fuera una binaria con cardinalidad (0,1)-(0,N) y tuviera atributos, deberíamos representarla en una mesa aparte. Así, el esquema de resolver la agregación el sería:

A(a0,a1)

CP:{a0}

B(b0, b1)

CP:{b0}

R(a0,b0)

CP:{b0}

CAj:{a0}->A

CAj:{ b0}->B

ES:{ a0}

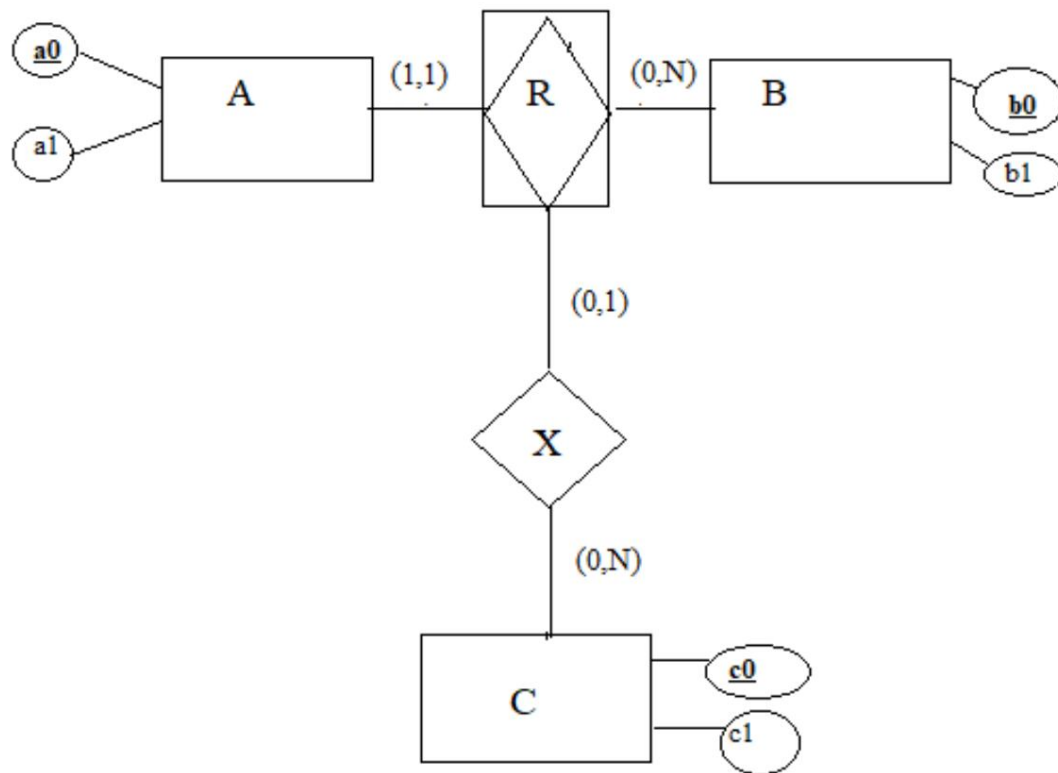
2. Ahora deberíamos resolver X siguiendo la lógica: la tabla donde está representada R es la que se debe de enlazar con la tabla derivada de la entidad C. Es una 1-N sin atributos:

C(c0, c1, b0)

CP:{c0}

CAj:{b0}->R

Caso 24



1. Si R (el objeto agregado) fuera una binaria con cardinalidad (1,1)-(0,N) y tuviera atributos, deberíamos representarla en la tabla B. Así, el esquema de resolver la agregación sería:

A(a0,a1)

CP:{a0}

B(b0, b1, a0)

CP:{b0}

CAj:{a0}->A

ES:{ a0}

2. Ahora deberíamos resolver X siguiendo la lógica de que la tabla donde está representada R es la que debe enlazarse con la derivada de la entidad C. Es una 1-N sin atributos:

C(c0, c1, b0)

CP:{c0}

CAj:{b0}->B